AUTOMATYKA • 2009 • Tom 13 • Zeszyt 3

Artur Sierszeń*

Cascade Algorithm for the Reference Set Size Reduction

1. Introduction

The conducted works consisting in the development of algorithms operating according to two different types of reduction:

1) incremental reduction – where the condensed set is constructed starting from an empty set which is then successively increased;
2) decremental (eliminative) reduction – where the condensed set is complete at first and then its elements are successively rejected until the stop criterion is met; have not clearly answered the question which of these approaches is more effective.

The analysis of very large sets indicated that incremental reduction leads to significant errors in the initial phase of operation; these errors require much time to be decreased. Eliminative reduction usually gives low classification error at the initial phase of computations; however, much time is needed to obtain considerable reduction.

2. Cascades algorithm

Combination of two considered types of the reference set condensation may consist of a sequence of both component algorithms. A model was implemented where first an incremental type of the algorithm based on reference set condensation was used and then, as the second phase, the modified Chang's method (Fig. 1) was applied, which is of the decremental type. The reverse sequence would require much more computations.

An important issue obtained by the author during the development of this algorithm was establishing of a transition criterion (stop criterion) from the first component algorithm to the second one. The transition criterion was based on the classification error (Fig. 2).

* Computer Engineering Department, Technical University of Łódź

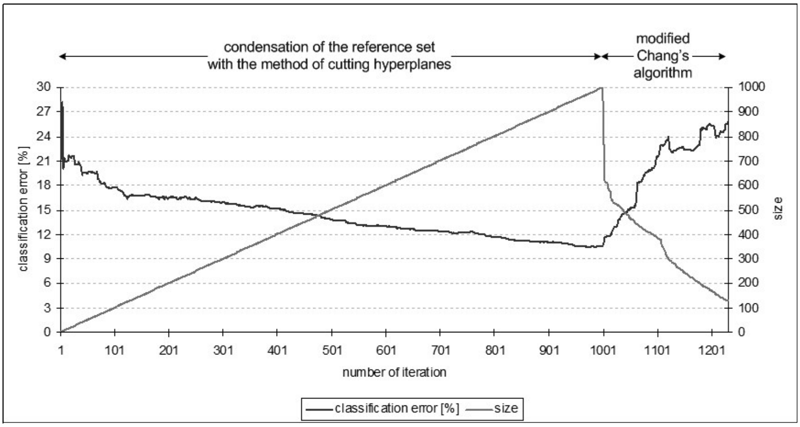

Fig. 1. Relationship between classification error and the size of the condensed set for the cascades algorithm

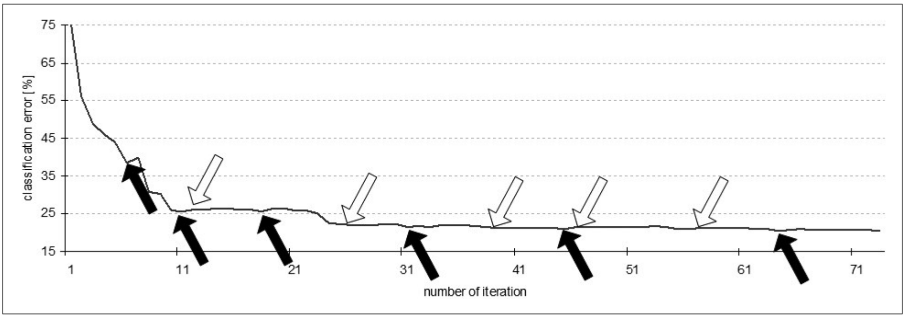

Fig. 2. Selection of the transition moment between the components of the cascades algorithm

Two intervals of classification error chart were analyzed:

1) periods of permanent unchanging of the error rate– the condensation algorithm does not significantly deteriorate the classification quality (in Fig. 2 – white arrows);
2) local minima – indicating that this stage (iteration) of reduction algorithm's operation improved the classification quality (in Fig. 2 – black arrows).

Both cases may decide whether at the specific moment transition to the second component algorithm should take place. In case of continuous and, for some time, relatively permanent or insignificantly growing classification error (this period, measured in a number of iterations, is determined by an operator/user of the algorithm), the transition criterion was recognized as a beginning of such a period. In case of a local minimum, it is important to decide in what period of time (defined by a number of iterations) this minimum should be determined. During the tests of the algorithm, in both cases the period was set to 1/1000 of a set's size was chosen, i.e. for a set of 7000 elements the period was constrained to 7 iterations was adopted. In a real application, it seems appropriate to determine this periods basing on the limit of time required to complete a given task (e.g. in quality control it is the time between the arrival of subsequent samples of the checked product on the assembly line). A criterion of error level may also be used, i.e. when the error reaches the chosen level, the transition will take place. One should also remember that the best solution seems to be an empirical selection of these parameters depending on a specific tasks and data set.

2.1. Reference set condensation rule based on the method of finding the mutually furthest points

The first component of the cascades algorithm is the condensation algorithm based on of finding the mutually furthest points. This method consists in assigning one pair of the mutually furthest points (from different classes) to each point from the learning set, assuming that in case of several furthest neighbours located at equal distances, the one with lowest number is always chosen. Because many objects may have the same pair of the mutually furthest points or a pair where a given point coincides with another, a pair with a lower number is chosen (i.e. a pair which was found as the first). The problem of coinciding points belonging to the same classes and having the same properties was solved by omitting them (removing them from test sets). It did not adversely affect the error level of classification performed with the use of obtained condensed sets. A pair of the mutually furthest points can be found for any subset of the primary reference set and is used to determine a hyperplane which divides this subset. It goes through the centre of the segment connecting these points and it is orthogonal to it. At the beginning the whole reference set is divided in this manner into two subsets. The next subset to be divided is determined automatically by the algorithm. New subsets obtained through division are replaced with gravity centres; they are assigned to a specific class by the majority criterion, i.e. they are assigned to the largest class inside the subset being currently divided. The figure below (Fig. 3) illustrates the operation (the first two iterations) of the algorithm for the example of 2-dimensional feature space.

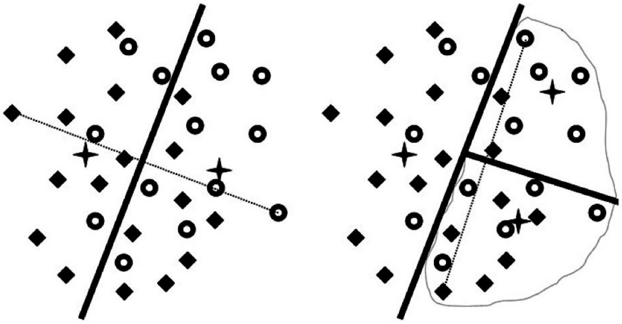

Fig. 3. The first two iterations of an exemplary operation of the presented algorithm (the first iteration – on the left, the second – on the right)

For the needs of the described condensation method, the modification of the algorithm of finding the mutually furthest points [1] was used. The operation of the algorithm is presented below with the use of a pseudocode.

```
T  –  the set of all testing objects, t – an element of T set;
t  =  [a1, a2 ... an]; a – a value of the feature describing the point; n – a number of
      features
i00.  START
i01.  choose tk = t0 (t0 = the first element of T set)
i02.  tz = tk
i03.  find tx element so that || tk , tx || = max, if tx = tz then go to i05
i04.  tz = tx, tk = tx, go to i03.
i05.  END
```

The graphic interpretation of the algorithm of finding the mutually furthest points was presented in the figure (Fig. 4). In each step of the algorithm, the distance to the furthest point from the opposite class is determined (Figs. 4a and 4b). In case this distance is the same as the one computed in the previous cycle, the algorithm returns a pair of mutually furthest points from different classes (Fig. 4c).

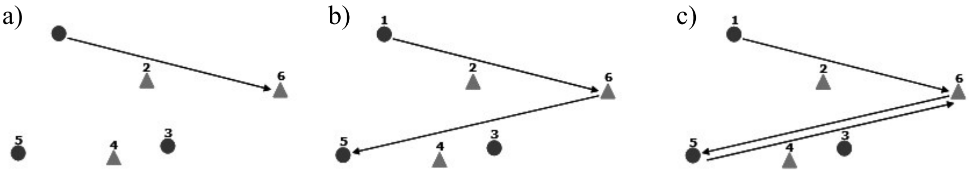

Fig. 4. The algorithm of finding the mutually furthest points: a) the first step; b) the second step; c) the third step)

Below, a pseudocode is presented for the reference set condensation algorithm based on determining the cutting hyperplanes.

```
T      the learning set containing m objects; Z – the condensed set;
Tj     subsets of the learning set, j=1,2,...,i, after performing ith iteration;
T  =   {t1, t2, ..., tm}; x, k – indexes of elements from the set T; m – number of elements
       in the primary reference set
t      = [a1, a2 ... an]; a – a feature describing an object; n – a number of features;
i00.   START; i=1; Ti = T; Z=∅ {i.e. null set}
i01.   Find a pair of the mutually furthest objects tj and tk in Ti set
i02.   Construct a cutting hyperplane of g(t)=0 equation basing on points tj and tk
i03.   TiA= {t∈Ti: g(t)>=0}; find a centre of gravity ziA of TiA set
i04.   TiB= {t∈Ti: g(t)<0}; find a centre of gravity ziB of TiB set
i05.   Delete Ti, remember TiA as Ti and TiB as Ti+1; next i=i+1
```

```
i06  Delete the gravity centre of Ti
i07. $Z= Z \cup \{ziA, ziB\}$
i08. Estimate classification error for 1-NN rule working with the condensed set Z and
     remember it
i09. Arrange Tj sets, $j=1,2,...i$, so that Ti is the largest set
i10. If Ti contains more than one objects, go to i01
i11. END
```

Each time after a new reduced set has been determined (i.e. after each iteration), classification error is computed, with the use of leave-one-out method, for 1-NN rule operating on the current condensed set.

2.2. Chang's algorithm

The second component algorithm is based on the modified Chang's algorithm. The original procedure of Chang's reduction [1] is presented below in the form of a pseudocode.

THE ORIGINAL CHANG'S ALGORITHM

```
T  =  $\{t1, t2, \dots, tm\}$ – the learning set containing m objects t;
t  =  $[a1, a2 \dots an]$; an element of T set, a – a feature describing a point; n – a number of features
Z  –  the current reduced set;
$\varnothing$  –  a null set; key1 and key2 – working variables of logic type;
i00. START, $A=\varnothing$, $B=T$; $A=\{$a random object from $B\}$;
i01. key1=false; key2=false; $B=B–A$;
i02. Find $p \in A$ and $q \in B$, so that the distance $d(p,q)$ is minimum;
i03. If p and q are from the same class, determine a set $Z=A \cup B \cup \{p^*\}–\{p,q\}$, where
     $p^*=(p+q)/2$;
i04. If p and q are from the same class and Z is the same as T, key1=true and
     key2=true;
i05. If key1=true, $A=A–\{p\} \cup \{p^*\}$ and $B=B–\{q\}$;
i06. If key1=false, $A=A \cup \{q\}$ and $B=B–\{q\}$;
i07. If $B \neq \varnothing$, go to i02;
i08. If $B=\varnothing$ and key2=true, $B=A$ and $A=\varnothing$ and go to i01;
i09. If $B=\varnothing$ and key2=false, $Z=A$ and END.
     'key1' is to register that two objects p and q have been replaced with one object
     p*, 'key2' is to recognise that no merger has taken place and that the algorithm
     should be ended.
```

The graphic illustration of the Chang's algorithm for the set of seven objects (five circles and two triangles) is presented in the figure (Fig. 5). The training set T of objects was presented in the figure (Fig. 5a). The algorithm starts from the complete reference set, i.e.

Z = T (Fig. 5b), which is the same as shown in Figure 5a. The nearest neighbour rule, operating on the reference set presented in Figure 5b, classifies correctly all objects from the learning set (i.e. objects from Fig. 5a). Since objects A and B lie closest to each other and they are from the same class, they are replaced with a new object H. The label of this object is the same as that of the objects A and B. All objects (from Fig. 5a) are still correctly classified, therefore, by replacing A and B with the prototype H, a new reduced set Z of objects is obtained; it is presented in Figure 5c. Analogically, after replacing H and C with object I, a status shown in Figure 5d is obtained. By merging F and G into object J, the next reduced set Z is obtained; it is shown in Figure 5e. By replacing D and E with object K, the status as presented in Figure 5f. is obtained. The reduced set Z presented in Figure 5f still correctly classifies the initial set (Fig. 5a). In case of merging I and J, some objects will be classified incorrectly. Thus, the process of merging and replacing objects is finished. Artificially created objects presented in Figure 5f will be used as a reference set for the nearest neighbour rule.

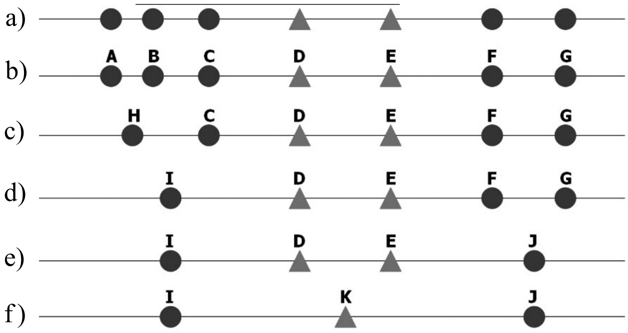

Fig. 5. Graphic interpretation of Chang's algorithm

The modification proposed by the author of this work aims at accelerating computations by replacing many objects, rather than a pair of them, with one object. For any object, all objects from the same class can be determined which are located at a shorter distance from it than any object from the opposite class. Then, all these objects are replaced with their gravity centre with the same label as that of the objects from which the gravity centre was computed.

The operation of the proposed modification of the Chang's algorithm with an example analogical to the one presented in Figure 5 is shown in Figure 6. During the first stage, a distance is found from a randomly selected point (in this case point A) to the nearest prototype representing the opposite class (D) (Fig. 6b). This distance, marked as x_1, enables finding all prototypes located closer than prototype D, that is B and C. After merging these points, a new point is obtained and marked as H (Fig. 6c). The new object is an average vector from the points which it replaced. The next randomly selected point may be point F. Again, the distance to the nearest object representing another class (E) is determined and marked as x_2. Then, points from the same class are sought for within a sphere of the radius x_2 (Fig. 6c). In this stage, only F and G are merged; as a result, the object J is created. Analogically, to the examples described above, a next point is selected at random again (this time it is prototype D), (Fig. 6d). The distance to the nearest prototype from another class is determined (prototype H) and marked as x_3. Then, all objects lying closer and belonging to the same class are found (only prototype E). The found prototype E is merged with D and they are replaced with prototype J.

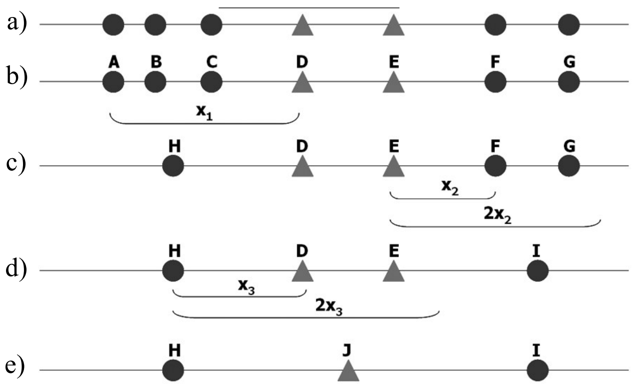

Fig. 6. Graphic illustration of modified Chang's algorithm

In each case, after a new set was determined, its reliability was checked by classification with 1-NN method and computation of the error rate with the leave-one-out method. The detailed description of 1-NN classifier's operation with a set reduced with the modified Chang's method is presented below.

MODIFIED CHANG'S ALGORITHM

```
T – the learning set containing m objects t;
Z – the current reduced set; P – the working set;
∅ – a null set; key1 and key2 – working variables of logic type;
i00.  START, $A=\varnothing$, $B=T$; $A=\{$a random object from $B\}$;
i01.  key1=false; key2=false; $B=B-A$;
i02.  Find $p\in A$ and $q\in B$ from other class than p object, so that the distance $d(p,q)$ is
      minimum;
i03.  Determine a set $P=\{t\in B\colon d(t,q)<d(p,q)\}$ and its centre of gravity $p^*$ with the
      same label as for p object;
i04.  Determine a set $Z=A\cup B\cup\{p^*\}-P$;
i05.  If Z does not worsen the classification of T set, key1=true and key2=true;
i06.  If key1=true, $A=A\cup\{p^*\}$ and $B=B-P$;
i07.  If key1=false, $A=A\cup P$ and $B=B-P$;
```

```
i08. If B≠∅, go to i02;
i09. If B=∅ and key2=true, B=A and A=∅ and go to i01;
i10. If B=∅ and key2=false, Z=A and END.
     'key1' is to register that two objects p and q have been replaced with one object
     p*, 'key2' is to recognise that no merger has taken place and that the algorithm
     should be ended.
```

3. Verification of the cascades algorithm

The Cascades algorithm for the reference set size reduction was implemented in C++ in Microsoft Visual Studio .NET 2003 environment. This allowed the author to test the method in Windows environment with the use of a PC computer equipped with Intel Pentium processor 4 HT 3GHz and 512MB of operating memory.

The computation tests were conducted with the use of sets from the repository of the University of California in Irvine (Machine Learning Repository, University of California, Irvine) [2]. These tests are commonly used in literature. These are the following (Tab. 1):

PHONEME – data set created as a result of an analysis of separate syllables pronunciation (e.g. pa, ta, pan etc.); what was taken into account in this analysis was the type of a vowel pronunciation – nasal or oral;

SATIMAGE – this data set was generated basing on the analysis of satellite pictures supported with other methods of observation (radar data, topographic maps, data concerning agriculture). Classes determine a kind of soil or a type of cultivation;

WAVEFORM – artificially generated data set, where each of the classes is created as a result of a combining 2 out of 3 sinusoids; for each attribute in a class a noise is generated.

Table 1
Parameters of the sets used during the tests

Name of the set	Number of classes	Number of features	Number of samples	Size of separate classes in the set					
				1	2	3	4	5	6
PHONEME	2	5	5404	3818	1586	–	–	–	–
SATIMAGE	6	36	6435	1533	703	1358	626	707	1508
WAVEFORM	3	21	5000	1657	1647	1696	–	–	–

All tests were repeated 25 times; the presented results (time of the algorithm's operation) were calculated as the average during each iteration. Computing the error with the leave-one-out method was definitely most time-consuming. Therefore, the author decided that the error should be computed every second iteration. This permitted to accelerate computations significantly.

3.1. *PHONEME* testing set

The charts present the results of the cascades algorithm operation for the *PHONEME* testing set. Apart from the original result obtained with the first stage of the cascade condensation algorithm (Fig. 7), four chosen examples are given which reflect the influence of selecting the point of transition between the component classifiers of the cascades algorithm (Figs. 8a, b, c, d).

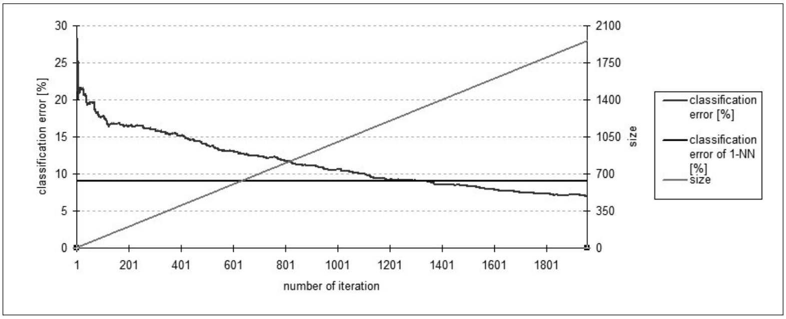

Fig. 7. The results of the first component algorithm of cascades condensation obtained with the use of the *PHONEME* testing set (without transition to the second component algorithm)

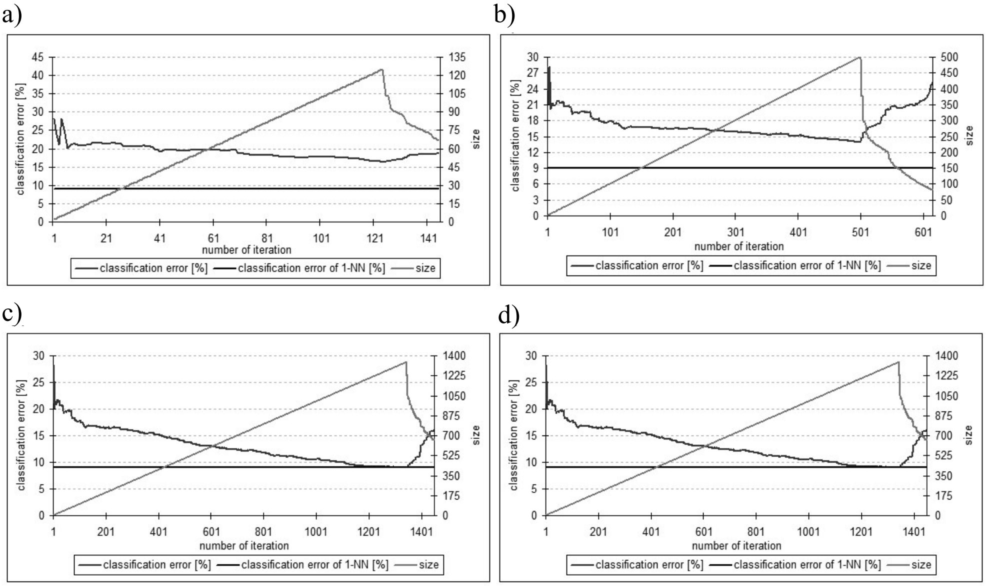

Fig. 8. Four chosen examples are given which reflect the influence of selecting the point of transition between the component classifiers (a – 124th iteration, b – 500th iteration, c – 1000th iteration, d – 1343rd iteration) obtained with the use of the PHONEME testing set

The analysis of results of the cascades algorithm operation on the PHONEME set shows the reduction of the reference set size. This process always increased the level of incorrect classifications; however, during the first few iterations after the transition, it could be always noticed that the error remained constant or slightly increased but the reference set size was considerably reduced.

In the first case (Fig. 8a), the first stage was finished after 124 iterations. The reference set contained 125 elements at that moment and the classification error amounted to 16.68%. The first three iterations of the second component algorithm reduced the reference set to 93 elements (i.e. 32 elements less, almost by 25%); the level of incorrect decisions increased to 16.81 at the same time (only 0.13% increase). In the second case (Fig. 8b), the first stage was finished after 500 iterations. The reference set contained 501 elements at that moment and the classification error reached the level of 13.93%. The first three iterations of the second component algorithm reduced the reference set to 374 elements (i.e. 126 less, almost 25%); the level of incorrect decisions increased to 14.71 at the same time (0.78% increase). In the third case (Fig. 8c), the first algorithm finished operating after 1000 iterations. The reference set contained 1001 elements at that moment and the classification error amounted to 10.57%. The first four iterations of the second component algorithm reduced the reference set to 619 elements (i.e. 382 elements less, over 38%); the level of incorrect decisions increased to 11.61 at the same time (1.04% increase). In the last examined case (Fig. 8d), the first algorithm finished operating after 1343 iterations. The reference set contained 1345 elements at that moment and the classification error amounted to 8.97% (it was equal to error of classification with 1-NN method). The first six iterations of the second component algorithm reduced the reference set to 1054 elements (i.e. 291 elements less, over 20%); the level of incorrect decisions increased to 9.09% at the same time (0.12% increase).

3.2. *SATIMAGE* testing set

The charts present the results of the cascades algorithm operation for the *SATIMAGE* testing set. Apart from the original result obtained with the first algorithm from the cascades solution (Fig. 9), four chosen examples are given which reflect the influence of selecting the point of transition between the component classifiers of the cascades algorithm (Figs. 10a, b, c, d).

The analysis of the results obtained for the SATIMAGE set with the use of cascade algorithm shows the reduction of the reference set size. This condensation always increased the level of incorrect classifications; however, during the first few iterations after the transition, it could be always noticed that the error remained constant or slightly increased but the reference set size was considerably reduced.

In the first case (Fig. 10a), the first stage was finished after 98 iterations. The reference set contained 99 elements at that moment and the classification error amounted to 18.25%. The first eight iterations of the second component algorithm reduced the reference set to 85 elements. It is little reduction, only 13 elements less, which amounts to 13%, but it was accompanied with improvement of classification quality. The level of incorrect decisions dropped to 17.97 (0.28% decrease). In the second case (Fig. 10b), the first stage was finished after 237 iterations. The reference set contained 238 elements at that moment and the classification error amounted to 14.59%. The first seven iterations of the second component algorithm reduced the reference set to 215 elements (i.e. 23 elements less, almost 10%); the level of incorrect decisions increased to 14.89% at the same time (0.3% increase). In the third case (Fig. 10c), the first stage was finished after 500 iterations. The reference set contained 501 elements at that moment and the classification error amounted to 12.76%.

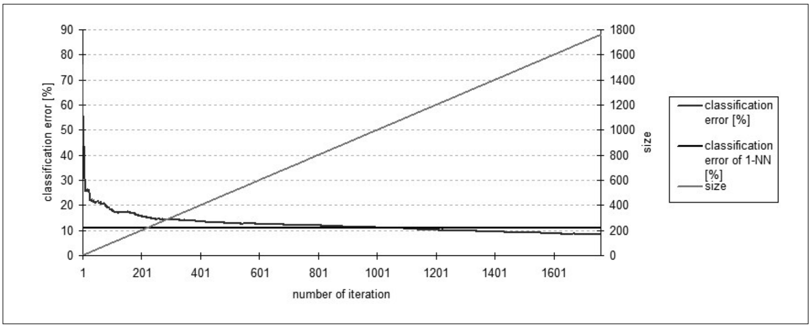

Fig. 9. The results of the first component algorithm of cascades condensation obtained with the use of the *SATIMAGE* testing set (without transition to the second component algorithm)

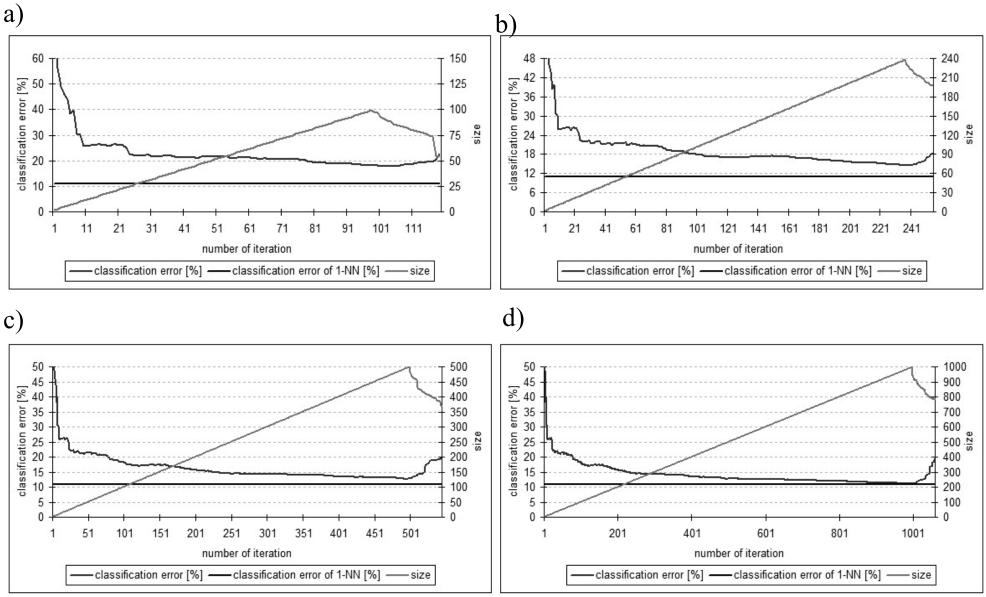

Fig. 10. Four chosen examples are given which reflect the influence of selecting the point of transition between the component classifiers (a – 98th iteration, b – 237th iteration, c – 500th iteration, d – 998th iteration) obtained with the use of the *SATIMAGE* testing set

The first seven iterations of the second component algorithm reduced the reference set to 460 elements (i.e. 41 elements less, almost 10%); the level of incorrect decisions increased to 13.50% (0.74% increase). In the last examined case (Fig. 10d), the first algorithm finished operating after 998 iterations. The reference set contained 999 elements at that moment and the classification error amounted to 11.15%. The first fourteen iterations of the second component algorithm reduced the reference set to 888 elements (i.e. 111 elements less, over 11%); the level of incorrect decisions increased to 11.80% (0.65% increase).

3.3. *WAVEFORM* testing set

The charts present the results of the cascade algorithm applied for the *WAVEFORM* testing set. Apart from the original result obtained with the first stage (Fig. 11), four chosen examples are given which reflect the influence of selecting the point of transition between the component classifiers of the cascades algorithm (Figs. 12a, b, c, d).

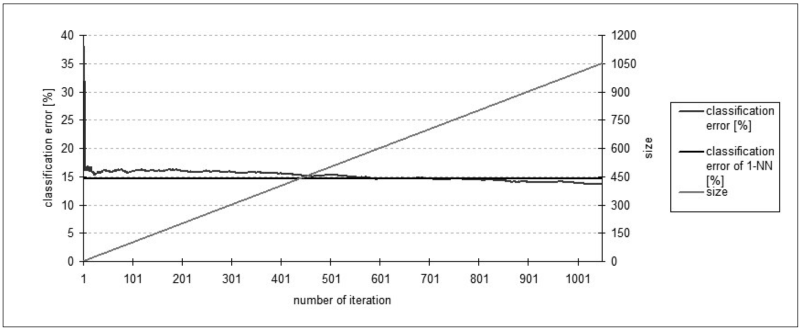

Fig. 11. The results of the first component algorithm of cascades condensation obtained with the use of the *WAVEFORM* testing set (without transition to the second component algorithm)

In the case presented in the first figure (Fig. 12a), the first algorithm stage was finished operating after 90 iterations. The reference set contained 91 elements at that moment and the classification error amounted to 15.64%. The first four iterations of the second component algorithm reduced the reference set to 79 elements (i.e. 11 elements less, over 12%); the classification error increased to 15.74 (0.1% increase). In the second case (Fig. 12b), the first algorithm finished operating after 215 iterations. The reference set contained 216 elements at that time and the classification error amounted to 15.83%. The first three iterations of the second component algorithm reduced the reference set to 211 elements (i.e. only 4 elements less); the number of incorrect decisions increased to 15.90% at the same time (0.07% increase). In the third case (Fig. 12c), the first stage was finished after 442 iterations. The reference set contained 443 elements at that moment and the classification error amounted to 15.21%. The first ten iterations of the second component algorithm reduced the reference set to 397 elements (i.e. as many as 46 elements less, over 10%); the number of incorrect decisions increased slightly to 15.23% at the same time (0.02% increase). In the last examined case (Fig. 12d), the first algorithm finished operating after 590 iterations. The reference set contained 591 elements at that moment and the classification error amounted to 14.68%. The first ten iterations of the second component algorithm reduced the reference set to 517 elements (i.e. 74 elements less, over 12%); the number of incorrect decisions increased to 15.23% at the same time (0.55% increase).

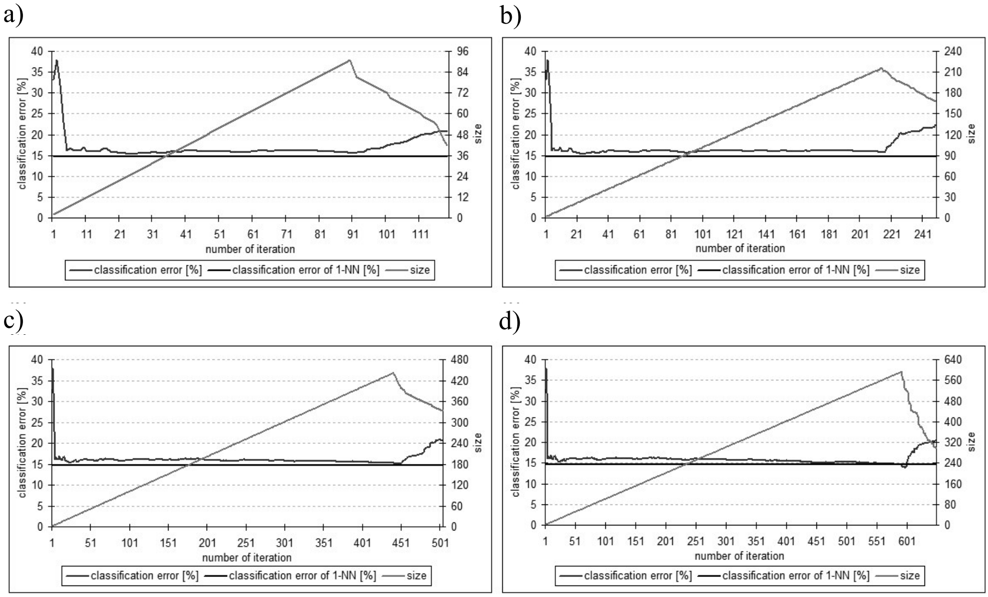

Fig. 12. Four chosen examples are given which reflect the influence of selecting the point of transition between the component classifiers (a – 90th iteration, b – 215th iteration, c – 442nd iteration, d – 590th iteration) obtained with the use of the *WAVEFORM* testing set

4. Conclusions

The presented results of experiments conducted with the use of three different testing sets indicated that the initial iterations of the proposed cascades algorithm give a considerable degree of condensation. This is accompanied by a slight increase in classification error or even sometimes by its decrease compared to the error for 1-NN method operating with the initial reference set. As it was already mentioned, the cascade algorithm combined from the same component stages in reverse order would require too much computations. The results of the conducted tests indicate that it is necessary to adjust experimentally the moment of transition from the cutting hyperplanes algorithm to the modified Chang's algorithm in case the cascades algorithm is used in real problems.

References

[1] Jóźwik A., Kieś P., *Reference set size reduction for 1-nn rule based on finding mutually nearest and mutually furthest pairs of points. Corres2005* – materiały konferencyjne, 2005.

[2] Merz C.H., Murphy P.M., *UCI repository of machine learning databases*. 1996, http://www.ics.uci.edu/~mlearn/MLRepository.html.